

DOCKET NO.: **GP-0002

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: **Hiang-Swee Chiang** Confirmation No.: **2184**
Serial No.: **09/810,716** Group Art Unit: 2193
Filing Date: **March 16, 2001** Examiner: **William H. Wood**
For: **WEB APPLICATION GENERATOR**

Mail Stop Appeal-Brief Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANT'S REPLY BRIEF PURSUANT TO 37 C.F.R. § 41.41

This Reply Brief is filed in response to Examiner's Answer mailed December 13, 2007, and in further support of Appellant's appeal from the rejections of claims 2, 4-29, 31-50, 52-64, 66-78, and 162-169. A Notice of Appeal was submitted on June 30, 2006 and an Appeal Brief was filed on January 31, 2007 (and resubmitted September 3, 2007).

The Office is hereby authorized to charge Deposit Account No. 23-3050 for any fee that may be due. The Commissioner is hereby requested to grant an extension of time for the appropriate length of time, should one be necessary, in connection with this filing or any future filing submitted to the U.S. Patent and Trademark Office in the above-identified application during the pendency of this application.

I. STATUS OF CLAIMS

The status of the claims is as follows:

Claim 1	Cancelled
Claims 2	Rejected and On Appeal
Claim 3	Cancelled
Claims 4-29	Rejected and On Appeal
Claim 30	Cancelled
Claims 31-50	Rejected and On Appeal
Claim 51	Cancelled
Claims 52-64	Rejected and On Appeal
Claim 65	Cancelled
Claims 66-78	Rejected and On Appeal
Claim 79-161	Cancelled
Claims 162-169	Rejected and On Appeal
Claims 170-174	Cancelled

II. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether the rejections of claims 2, 4-29, 31-50, 52-64, 66-78, and 162-169 as unpatentable under 35 U.S.C. §103(a) over Lau (U.S. Patent No. 5,987,247) in view of Lindhorst et al. (U.S. Patent No. 6,337,696) in further view of Quaeler-Bock et al.(U.S. Patent No. 6,023,271) is proper.

III. ARGUMENT

Appellant respectfully submits that the Examiner's Answer fails to establish that all of the recited claim language is taught by the cited references.

A. Overview Of The Disclosed Embodiments

Appellant's application discloses methods and systems for generating a web application. In an exemplary method, a web application server receives input files from graphic designers or business analysts. The input files may comprise at least one web application graphical user interface. The web application server determines if an application framework code is available for the web application, and retrieves the application framework code from an application directory. If the application framework code is not available for the web application, then the web application server generates the application framework code, along with a business logic foundation code, an event handler skeleton and a graphical user interface code. Generating the event handler skeleton comprises parsing the input files, reviewing each parsed input file for a tag type, attribute name and attribute value, and determining an event handler method based on the tag type, attribute name and attribute value.

In particular, in connection with generating the event handler skeleton, the application specification explains as follows:

After identifying the input tag, web application generator 205 searches for specific attributes within the input tag, based on the input tag type. In one embodiment, certain tags have a single attribute, while other tags have multiple attributes. After identifying each attribute within the tag, web application generator 205 identifies an attribute value associated with the attribute name. *Ultimately, **based on the input tag, attribute name(s) and associated attribute value(s)**, web application generator 205 relies on a particular "rule" (or fixed formula) within web application server memory 325 to generate event handler code 620 and GUI code 625 (step 730). Thus, the input tag, attribute name and attribute value determine the "rule" that web application generator 205 relies on to generate event handler code 620 and GUI code 625.*

For example, in one embodiment, one tag of an Input file in HTML format is "<input type=submit name=JLMClick

*value=Add>“. Web application generator 205 first identifies the tag as an “input” tag, next identifies the attribute names as “name” and “value”, and further identifies the associated attribute values as “JMLClick” and “Add”. **Based on the input tag, attribute names and associated attribute values, web application generator 205 relies on a particular rule in web application server memory 325 to generate the following event handler method: “public void add (App app) throws Exception”, as well as a GUI code in the form of a source file such as “indexClick.java.” If one of the attribute values was something other than JLMClick, then web application generator 205 would have relied on a separate rule in web application server memory 325 to generate event handler code 620 and GUI code 625.** (Application Specification, p. 15)*

The web application server receives web application business logic objects and event handlers from the web developers, and organizes the application framework code, web application business logic objects and event handler methods into web application source code. The web application server then compiles the web application source code. Modified input files may subsequently be received by the web application server from the graphic designers or business analysts. The modified input files are compiled and dynamically bound with the compiled web application source code at runtime.

B. The Claim Language

Representative claim 2 recites as follows:

A method of generating computer code for a web application, comprising:
 receiving input files, wherein the input files are at least one web application graphical user interface;
generating an application framework code and an event handler skeleton, wherein generating an event handler skeleton comprises:
 parsing at least one input file;
 reviewing the parsed input file for one or more of a tag type, an attribute name and an attribute value; and
 determining an event handler method based on one or more of the tag type, the attribute name and the attribute value;
 receiving web application business logic objects;
 receiving event handler methods;

organizing the application framework code, the web application business logic objects and the event handler methods into application source code; and
binding the web application source code with the input files at runtime.

C. The References Do Not Teach The Recited Claim Language

It is respectfully submitted that the Examiner's Answer fails to demonstrate where in the cited references the recited claim language is present. In particular, the Answer fails to illustrate that Lindhorst discloses "generating an event handler skeleton compris[ing] . . . determining an event handler method **based on one or more of the tag type, the attribute name and attribute value.**" Applicant respectfully submits that Lindhorst does not teach the recited language and cannot possibly suggest the recited combination. *See* M.P.E.P. § 2143.03 ("[t]o establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art.").

The Examiner's Answer alleges that "the system of Lindhorst is very much involved in the 'determining an event handler method based on one or more of the tag type, the attribute name and the attribute value.'" (Answer, p. 15-16). The Answer further asserts "[t]his is clear from 'outputs events to the event panel corresponding to events that are associated with scriptable HTML tags' (column 13, lines 22-24) and the subsequent Table 2 (column 13, lines 29-56)." Appellant respectfully disagrees.

Lindhorst teaches a software application wherein events are displayed on an event pane of a user interface (col. 13, ll. 22-24) and actions are displayed on an action pane of the user interface (col. 13, ll. 65-67). Lindhorst teaches that a **user selects** an event icon in the event pane in order to link that event to a desired action in the action pane. (Abstract). A third panel displays the link established by the user's selected event and action. (Col. 3, ll. 24-25).

Thus, in the systems and methods described by Lindhorst, the **user selects** events to be linked with actions. Therefore, the links displayed in the third panel of the interface disclosed by Lindhorst are "**based on** **the user inputs** and **not** **based on one or more of the tag type, the attribute name and attribute value.**"

The Answer argues that "[a] user is involved in the process but that involvement is 'based on' certain tags having certain/appropriate events and methods." (Answer, p. 16).

Importantly, the Answer provides no support from Lindhorst for this assertion. Furthermore, even if such support existed, the claim recites “**determining an event handler method based on** one or more of the tag type, the attribute name and attribute value.” The claim does **not** recite “determining **a user’s involvement in the process** based on one or more of the tag type, the attribute name and attribute value.”

The Answer notes that “Table 2’s ‘HTML tag’ demonstrates tag type . . . , attribute name . . . and attribute value.” (Answer, p. 16). But Lindhorst does ***not*** indicate that the listed tags, events, or methods are employed by the user to “determine[e] an event handler method.” Rather, as Lindhorst itself acknowledges, Table 2 merely provides a listing of exemplary scriptable tags as defined in the HTML 3.0 specification. (Col. 13, ll. 24-26). The listed tags are those that are defined in the HTML specification as being scriptable (as compared to those tags that are not scriptable). Thus, Table 2 of Lindhorst does not teach or suggest that the “user” described in Lindhorst makes a determination “**based on** one or more of the tag type, the attribute name and attribute value.”

The Answer suggests that the claims are entitled to “the broadest reasonable interpretation.” (See Answer, p. 16-17). This may be true. But Appellant respectfully notes that an essential word in that standard is “reasonable” – “the broadest **reasonable** interpretation.” It is respectfully submitted that the interpretations being alleged by the Examiner are *not reasonable*. In fact, the Answer’s alleged interpretation would essentially read out the actual language of the claim. For example, the Answer alleges that “[t]he claimed terminology is of sufficient breadth that ‘determining . . . based on’ is proven so long as the elements are present in ***any*** point in the process.” (Answer, p. 15). Under this interpretation, there would be no limit to what “determining an event handler” would be “based on.” For example, if there happened to be a program counter in a program that was present in process of arriving at an event handler, then the determining an event handler would be “based on” the program counter as it is “present at any point in the process.”

The Answer alleges that the phrase “based on” does not confer that a user must choose ‘solely based on’ nor does it even indicate that a user must select a course of action in which the “tag type, attribute name and attribute value’ have the most important, final or direct impact.” (Answer, p. 17). Thus, according to this interpretation, even if an event handler method were selected using a random number generator, the determining the event

handler method would still also be “based on” the program counter that is used in a program that generates the random number.

Respectfully, the claim does not recite “determining an event handler method wherein a tag type, attribute name, or attribute value is present at any point in the process” or any other contrived language. Rather, the actual language recited in the claim is “determining an event handler method **based on one or more of the tag type, the attribute name and the attribute value.**” It is respectfully submitted that the proposed interpretations are not “reasonable” and run afoul of the actual language of the claim.

In the systems and methods described by Lindhorst, the *user selects* events to be linked with actions. The links displayed in the third panel of the interface disclosed by Lindhorst are “**based on**” **the user inputs.** In contrast to claim 2, Lindhorst does not disclose “determining an event handler method **based on one or more of the tag type, the attribute name and attribute value.**”

Therefore, because neither Lau nor Lindhorst teach the recited claim language, the cited references do not render obvious claim 2 and all claims depending therefrom. *See* M.P.E.P. § 2143.03. For similar reasons, independent claims 27, 48, 64, 78, 162, 163, 164, 165, and 166, and all claims depending therefrom are patentable over the cited references. Withdrawal of the rejections under 35 U.S.C. § 103(a) is respectfully requested.

Date: February 13, 2008

/John E. McGlynn/
John E. McGlynn
Registration No. 42,863

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439